(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
21 December 2000 (21.12.2000)

PCT

(10) International Publication Number
WO 00/77984 A1

(51) International Patent Classification⁷: H04L 12/56, G06F 17/30

(21) International Application Number: PCT/SE00/01162

(22) International Filing Date: 6 June 2000 (06.06.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
9902175-0 10 June 1999 (10.06.1999) SE

(71) Applicant: TELEFONAKTIEBOLAGET L M ERICSSON (publ) [SE/SE]; S-126 25 Stockholm (SE).

(72) Inventor: STRÖMBERGSON, Joachim; Axgatan 6, S-431 40 Mölndal (SE).

(74) Agents: BERGENTALL, Annika et al.; Cegumark AB, P.O. Box 53047, S-400 14 Göteborg (SE).

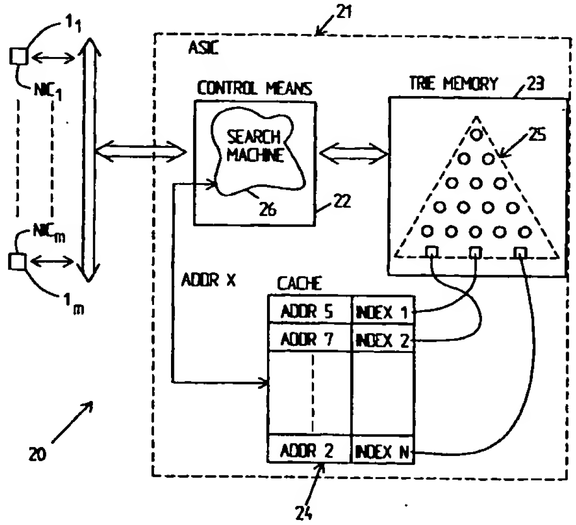
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *With international search report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: ARRANGEMENT, SYSTEM AND METHOD RELATING TO DATACOMMUNICATION



(57) **Abstract:** The present invention relates to an arrangement (20), system and a method respectively for forwarding data packets through a routing device including router storing means (23) including a digital search tree, a trie (25), the searching of which comprises traversion the tree from the root to a leaf through a number of nodes. In addition to the trie (25) a cache memory (24) is provided for storing of a number of indices to the leaves containing next-hop addresses. For looking-up a next-hop address for an incoming packet, a search is performed at least in the cache memory (24), and in absence of a cache hit, a search is also performed in the trie (25). This can be done in parallel or in series.

Title:

ARRANGEMENT, SYSTEM AND METHOD RELATING TO DATACOMMUNICATION

TECHNICAL FIELD

The present invention relates to data routing and particularly to an arrangement, a system and a method for forwarding data packets through a routing device.

STATE OF THE ART

In packet switched communication a routing device is considered to be a switching device receiving packets on input ports, and using destination information associated with the packet, routing the packets to the relevant destination or an intermediary destination via an appropriate output port. Due particularly to the rapid development of the Internet it has become exceedingly important to be able to utilize available network capacity as efficiently as possible. Due to the development within transmission technology the link capacity is high. A limiting factor has however become the routing devices within or between networks and it has been advised to use switching as much as possible instead of routing. A router may be connected to a plurality of networks and forward traffic within and between such networks which means that through such nodes comprising a routing device, an enormous amount of data traffic passes. A routing device determines how every packet should be sent on. A router has to keep track of the best way for forwarding a packet. Therefore it needs to have address information and status of the neighbouring routing devices. Such information is stored in routing tables. Address look-up is a central function of a routing device. It is well known to use a central processor serving all the input ports of a routing device to perform address look-up. A header of a packet is then received in the processor, also called a forwarding engine, which uses the destination address to determine the output port relevant for the packet and its next-hop address.

In a routing device in which the next-hop information is stored in a conventional table, there has to be some kind of mapping between an incoming IP-address and the memory address or the index where its next-hop information is stored. In order to perform an address look-up, the IP-address is transformed to a correct memory address which then is followed by a look-up in the table.

For speeding up the look-up it is known to use so called digital search trees as storing means in many state of the art devices. Such search trees are also called tries (cf. retrieval). In a trie based routing device there is no such corresponding mapping as referred to above between an IP-address and the location for storing of the next-hop information. Instead an analysis is performed of the incoming IP-address in several steps. Each step gives a new node in the digital search tree in which leafs correspond to records in a file. Searching proceeds from the root to a leaf. The (each) node is then analysed so that it can be established which of its children that is the next step towards the end node (cf. leaf as referred to above) containing the next-hop information. This means that the tree is traversed node by node and in every step the information has to be fetched. The consequence thereof is that a look-up to find next-hop information involves a traversal of the digital search tree which technically means that several memory accesses have to be done. For routing devices for which the storing means exclusively consist of tables, the next-hop tables will be very large. To be able to perform a fast look-up such tables are stored in fast and expensive memories. In order to still further increase the performance of a routing device, the memories are provided with even faster, fully associative cache memories so that a copy of a recently found next-hop information can be stored therein. A cache memory is a high-speed memory which stores data that the computer can access very quickly. It may be an independent high-speed storing device or it may constitute a particular section of a main memory. This functions since there is a temporary location connection between the memory access or in other words, if the next-hop information for an address has been looked-up, the probability is high that the same information will be needed again within the nearest future. The reason is that an IP-packet rarely is sent alone but as a part of a flow of several packets.

In routing devices in which the storing means are implemented as a digital search tree, or a trie, as referred to above, the table is represented through a search tree. The search tree itself can be compressed in different manners and it can be made so small that there will be room for it in the internal data cache of a commercial processor. The result thereof is that a processor can do next-hop look-ups rapidly. However, all memory accesser to find the node with the next-hop information have to be performed. Each such memory access is however demanding and since a large number of memory accesses may be needed for each address, this severely limits the performance of a routing device. Various trie based routing devices are known, see for example ”Small Forwarding Tables for Fast Routing Lookups”, by Mikael Degermark et al, Research Report 1997:02, Luleå University of Technology, Department of Computer Science and Electrical Engineering, Division of Computer Communication and ”Fast IP Routing with LC-Tries”, by Stefan Nilsson, Gunnar Karlsson, Dr Dobb’s Journal, August 1998, pp. 70-75.

In general various ways of using compression technique algorithms applied on routing, are known. However, none of them provides a satisfactory solution enabling a fast and efficient look-up of next-hop information.

SUMMARY OF THE INVENTION

What is needed is therefore an arrangement for forwarding data packets through a routing device having a high performance. Particularly an arrangement is needed through which the look-up of next-hop addresses can be performed in an efficient and fast manner, particularly so that the number of next-hop addresses that can be found per time unit is high and particularly higher than with hitherto known arrangements. An arrangement is also needed through which a packet can be transported in a fast and safe manner through a node in an IP-based network. Particularly an arrangement is needed through which the number of memory accesses can be reduced, thus increasing the performance. A trie based routing device is also needed which fulfills the above mentioned objects. Particularly such an arrangement is needed which is cheap and easy to build.

A method of forwarding data packets through a routing device is also needed through which the above mentioned objects can be fulfilled. Still further a communication system or a network of communication systems including a number of routing devices through which the above mentioned objects are fulfilled is needed.

Therefore an arrangement for forwarding data packets within a packet switched network or between networks, through a routing device comprising a number of input and output ports respectively and a forwarding engine comprising router storing means including a digital search tree, also called a trie, is needed. The searching in such a trie comprises traversing the tree from the route to a leaf through a number of nodes. Control means, e.g. a forwarding engine are provided including a search machine for controlling address look up for incoming packets.

A cache memory is provided in addition to the digital search tree. In the cache a number of indices to the leafs of the trie for a number of next-hop addresses is provided. For looking up a next-hop address for an incoming packet, a search is performed at least in the cache memory. The control means are used for receiving incoming addresses, e.g. IP-addresses and for initiating a look-up of the corresponding next-hop address which comprises a look-up in the cache memory and a traversion of the digital search tree (if needed).

Particularly a traversion of a digital search tree is initiated substantially simultaneously with the look-up in the cache memory. Alternatively the traversion of the digital search tree is only initiated if the look-up in the cache memory is unsuccessful, i.e. if the address with an index to the next-hop address can not be found. This means that either the search in the digital search tree is done in parallel with the look-up in the cache memory or it is initiated in case of no cache hit. If the search in the trie and the cache look-up are performed substantially in parallell, the traversion of the search tree is interrupted as soon as there is a cache hit.

Particularly indices to the leaves of a number of the most recently looked-up next-hop addresses are stored in the cache memory. The index cache is advantageously formed in such a manner that the least recently used addresses in the cache successively are replaced by more recently looked-up addresses. In an alternative embodiment a time interval is defined for replacement of addresses in the cache memory such that addresses not having been used during such interval are replaced by new addresses. This can be done in different manners, using various priority requirements etc.

Particularly the access time for a cache access is shorter than a digital search tree memory access. The cache may be a direct map cache, a set associative cache or a fully associative cache.

In a particularly advantageous embodiment the arrangement is implemented as a FPGA (Field Programmable Gate Array) chip. In an alternative implementation the arrangement is implemented as an ASIC (Application Specification Integrated Circuit) chip. The storing means, i.e. the digital search tree, may be implemented as a digital tree structure which is stored on-chip. Alternatively it is stored off-chip.

Particularly the cache memory is provided on-chip.

In a particular implementation the arrangement comprises an IP-router, i.e. an Internet router. Particularly the routing table comprising a digital search tree comprises a compressed data structure. The digital search tree may comprise an LC-trie, a Patricia trie or any other convenient trie known in the art.

In a most advantageous embodiment a number of addresses are looked-up in parallell. This can also be implemented through some kind of pipe-lining.

Therefore also a method for forwarding packets through a routing device of a packet switched data network is provided. The routing device includes control means (a forwarding engine) and router storing means comprising a digital search tree. The method includes the steps of storing indices to the next-hop addresses for a number of data packet addresses in a cache memory; using the cache memory and a traversion of the digital search tree to find an index to the next-hop address or the leaf containing the next-hop address respectively; interrupting the traversion of the digital search tree in case of a cache hit, and otherwise proceeding with the traversion of the digital search tree until the next-hop address is found. Thereon follow the steps of fetching the next-hop address information and forwarding the packet to the relevant output port using the found next-hop information.

The method particularly includes the steps of first performing a look-up in the cache memory and, in case of no cache hit, initiating a traversion of the digital search tree. In an alternative implementation the method includes the steps of simultaneously or parallelly performing a look-up in the cache memory and initiating a traversion of the digital search tree and interrupting the traversion of the digital search tree in case of a cache hit and otherwise proceeding with the traversion until the leaf containing the next-hop address is found in the search tree. The method includes particularly the step of, when a next-hop address is found through the traversion of the digital search tree, storing the index to said next-hop address into the cache memory, e.g. replacing the least recently accessed address information if there is no free space in the cache by the index to the most recently used address information.

According to different embodiments the cache memory is e.g. a direct map cache, a set associative or a fully associative index cache provided on-chip of an FPGA or an ASIC used for the implementation of the routing device. Still further the method is characterized through the step of looking up a number of next-hop addresses substantially in parallell (or according to a pipe-line procedure).

A packet data communication system is also provided which comprises a number of routing devices within a network or for interconnecting networks, each routing device comprising a number of input and output ports respectively, control means with a number of forwarding engines, router storing means including a digital search tree or a trie containing next-hop address information. At least a number of the routing devices comprise a cache memory for storing indices to a number of next-hop addresses contained in the digital search tree. For finding a next-hop address, a look-up is performed at least in the cache memory.

Particularly a traversion of the digital search tree is initiated substantially simultaneously with the look-up in the cache memory, which traversion is interrupted upon a cache hit. Alternatively a traversion of the digital search tree is only initiated if there is no cache hit.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention will in the following be further described, in a non-limiting way, and with reference to the accompanying drawings in which:

FIG 1 is a simplified, schematical illustration of an arrangement according to the invention,

FIG 2 shows a storing of nodes/a leaf into a memory,

FIG 3 is a simplified illustration of a trie, FIG 4 illustrates more in detail an embodiment of an arrangement according to invention,

FIG 5 shows another embodiment of an arrangement according to the invention,

FIG 6 is a flow diagram schematically illustrating the look-up of a next-hop address according to the invention,

FIG 7 is a flow diagram illustrating access to the trie memory, and

FIG 8 is a flow diagram illustrating the finding of a next-hop address using a trie and a cache according to the invention.

DETAILED DESCRIPTION OF THE INVENTION

Fig. 1 describes an arrangement according to the invention in a simplified manner. According to the invention the temporary localisation between packets is considered even if a trie based routing device is used. It is here supposed that packets are input to control means 2. In this particular embodiment it is supposed that the packets are so called IP-packets (of the Internet) having an IP-address. The look-up of the next-hop information corresponding to the IP-address of an incoming packet is controlled by the control means 2 which, in this embodiment, simultaneously initiates a traversion of the digital search tree (the trie) 5 of the router storing means 3 and a look-up in the index cache 4. If one of the rows in the cache 4 contains the relevant IP-address, there has already been a successful look-up in the trie 5 (recently). It is then possible to interrupt the traversion of the trie 5. Using the index corresponding to the relevant IP-address in the cache memory 4, the information contained in the leaf can be directly fetched from the trie 5.

If however there is no such row corresponding to the relevant IP-address the traversion of the trie 5 proceeds. The traversion is ended when the leaf has been found. The index to the leaf (contained in the parent node of a leaf) is stored together with the incoming IP-address into the cache 4, preferably through replacing the least recently used row (address) in the cache.

A successful traversion of the digital search tree requires at least one memory access. Since the memory, here the router storing means 3, is supposed to be much slower than the cache, a single trie memory access requires more cycles than a look-up in the cache 4. A successful look-up in the cache 4 has therefore as a consequence that the total time as counted in cycles, to find the wanted next-hop information, is reduced considerably. Generally it is not sufficient with one memory access to find a leaf in the tree but it requires several memory accesses.

Fig. 2 very schematically illustrates a digital search tree or a trie 5′ with a number of nodes 1,2,3,4 and a leaf 5. The figure also illustrates that the nodes and the leaf of the trie 5′ can be stored in the cache 4′ in a manner that is independent of the location, i.e. node 1 stored at the top followed by node 3 whereas the leaf is stored in the middle followed by nodes 4 and 2 respectively corresponding to memory locations $m_1,\ldots,m_n$.

Fig. 3 is a schematical illustration of how a trie can be traversed. "Trie" originates from retrieval and it comprises a binary digital search tree in which the bit pattern of the addresses is traversed and in which one or more bits give(s) which child of a given node that should be selected in order to reach the final node, or the leaf. Fig. 3 shows how starting from the top of the tree, through examining the bits contained in the IP-address from the left to right in every node, either route 0 or route 1 downwards in the tree is selected. When a node having no sub-nodes has been reached, also denoted a leaf, it is checked whether the rest of the IP-address is contained in the leaf. If this is the case, the route to the correct leaf has been found. Here it is supposed that the IP-address is 0010.

Fig. 4 shows one embodiment of the invention comprising an arrangement 20 in which packets are input over network interfaces $NIC_1,\ldots,NIC_m$, $1_1,\ldots,1_m$ corresponding to input ports with the same number. An input packet with address x is via databuses input to controlling means 22 comprising a search machine 26A look-up to find the next-hop address and output port (output NIC) is requested. The search machine 26 then performs a look-up in the cache memory 24 and in the trie memory 23 containing the digital search tree 25. This can be done in series or in parallell. Recently looked-up addresses-indices are located at different locations in the cache memory 24 and they need not be sorted as already discussed above. When the relevant information, i.e. the next-hop address and the output port, has been found, either in the cache 24 or in the trie 25, and fetched, the packet is sent out on the appropriate NIC. As already discussed above, if a look-up has been initiated in the trie 25, this is interrupted if there is a cache hit, i.e. if the address with the accompanying index to a leaf in the trie has been found in the cache. Alternatively the cache 24 is searched first and, if there is a cache hit, the traversion of the trie 25 need not even be initiated. If however there is no cache hit but the relevant information is contained only in the trie 25, the index to the leaf in the trie is stored into the cache advantageously replacing the least recently accessed information or address in the cache. Any other appropriate criteria may however also be used.

In this embodiment the arrangement is implemented in hardware comprising an ASIC 21. In this case the trie memory 23 is provided on-chip. However, alternatively it could also have been provided off-chip. However, the cache is implemented as a fast, fully associative cache on-chip in the ASIC and the access time for a look-up in the cache is much shorter than the time it takes for a memory access of the trie 25. In alternative embodiments the cache comprises a direct map cache or a set associative cache.

Fig. 5 shows another embodiment of the invention comprising an arrangement 30 in which, similar to Fig. 4, packets arrive over network interfaces 1_1-1_m which then over databuses are input to control means 32. Also here a control means 32 comprises a search machine 26 (or a forwarding engine) responsible for the look-ups in the trie memory 33 and in the cache memory 34. Also in this case the look-ups can be done parallelly or in series. Like in the preceding figure the trie memory 33 comprises a trie 35. However, in this case the arrangement is implemented as a FPGA (Field Programmable Gate Array) containing the control means 32 and the cache memory 34. The trie memory 33 is here provided off-chip, i.e. is not contained on the chip.

In an alternative implementation, not shown, the FPGA chip also contains the trie memory. However, through keeping the trie memory 33 outside the chip 31, a small and cheap on-chip implementation is enabled. In other aspects the functioning is the same as that discussed with reference to Fig. 4.

In Fig. 6, which is a simplified flow diagram describing one exemplary flow, it is supposed that a packet (e.g. an IP-packet) is received in a routing device, 1010. The IP-address is then extracted, 1020, and a cache look-up is performed, 1030A, substantially at the same time as a traversion of the trie is initiated, starting with node 1, 1030B.

After, or even before, the access of the first node is completed it is settled whether the address was found in the cache, 1040A. If yes, the index corresponding to the found address is used to find the leaf in the trie, 1050A. The next-hop information and information about output port is then fetched from the trie leaf, 1060A. Then the packet is forwarded to the relevant output port, 1070A. If, however, there was no cache hit, the traversion of the trie proceeds with node n = n + 1; n = 1,...,p 1040B, until the leaf is found (p here simply denotes the leaf). The wanted information is then fetched, 1050B, the packet is sent on, 1060B, as above, and the address with the index to the leaf is stored into the cache, replacing the least recently used row, 1070B.

Fig. 7 is a flow diagram illustrating a traversion of a trie. In the figure is not illustrated any preceding or parallell look-up in the cache memory. The flow is supposed to start, 100, with the input of a packet to control means. The, in this case, IP-address is then provided to the top node of the digital tree structure, 110. The node is then fetched, 120, and the node is checked for a match, i.e. it is examined whether the searched address is the same as the address in the node, 130. If there is a match, i.e. if the searched address corresponds to the address in the node, the searched information is fetched from the node, 131, and, according to the invention, the address index is stored into the cache memory, 132. Particularly the address index is provided in the node which is a parent to the leaf in the tree structure. The procedure then comes to an end, 133, although it is of course proceeded with sending the packet along with the relevant next-hop address information to the relevant output port etc.

If however it was established that the searched address did not correspond to the address in the node, it is examined whether the searched address is lower than the address in the node, 140. if yes, the address to the left child node is fetched from the node, 141 and the left child node is fetched, 120 etc. If however it was established that the searched address was not smaller than the address in the node, it is established whether the searched address is higher than the address in the node, 150. If yes, the address to the right child node is fetched from the node, 151, and the right child node is fetched, 120, and the procedure goes on as discussed above.

A memory access to the trie memory corresponds to step 120. Through the implementation of a cache memory according to the invention, the number of such accesses are reduced considerably.

If however the whole tree is traversed without finding any matching address in any node, a new tree has to be built and this is done in a conventional manner. However, when there is a change of trees in the trie memory, the cache has to be emptied. The index already provided in a cache runs the risk of being wrong.

Fig. 8 is a flow diagram describing address look-up using a cache in addition to the trie memory. Again is supposed that the start, 200, includes the input of an address, e.g. IP-address, to the control means of an arrangement according to the invention. The input address is then matched with all addresses contained in the cache memory, 210, to see if there is a cache hit, 220. If yes, the searched node is fetched from the trie memory, 221, and the searched information is extracted from the node, 222, whereupon the procedure, as far as memory access is concerned, comes to an end, 223. If however it was established that there was no cache hit, a normal trie memory access is performed, 230, as discussed with reference to Fig. 7. However, Fig. 7 shows one particular implementation. It is also possible to build a search machine that runs both the flows simultaneously. In case of a cache hit, the normal flow is interrupted. In case of a cache miss, the flow as described above is interrupted since a normal flow already is running. However, when the trie memory access 131, has been performed (successfully), the index is stored into the cache, 231, whereupon the memory access flow comes to an end, 232.

In an alternative implementation the steps containing fetching a node, extraction of information and cache look-up can be divided into three pipe-lined steps and a new look-up can be initiated every cycle.

Parallell look-ups of more than one address can also be performed. Also other ways of performing a pipe-lining procedure can be implemented. A single flow may also be pipe-lined in an alternative manner.

Still further the inventive concept can be used in combination with any kind of compressed algorithms etc. to speed up the traversion of the trie based structure.

The inventive concept can be implemented to provide Gbyte routers as well as Tbyte routers and this is implementable for IPv.4 as well as IPv.6; the principle remains the same. Also in other aspects the invention is not limited to the specifically illustrated embodiments but can be varied in a number of ways within the scope of the appended claims.

CLAIMS

1. An arrangement (10;20;30) for forwarding data packets through a routing device in a packet switched network or between networks, said routing device comprising a number of input ports, a number of output ports and control means (2) such as a forwarding engine for controlling address look up and forwarding of packets received at input ports, router storing means (3;23;33)including a digital search tree (trie) (5;5';25;35), the searching of which comprises traversing the tree from the root to a leaf through a number of nodes,

characterized in

that a cache memory (4;4';24;34) is provided for storing of a number of addresses with corresponding indices to the leafs of a number of next-hop addresses and in that for looking up a next-hop address for an incoming packet, a search is performed at least in the cache memory (4;4';24;34).

2. An arrangement according to claim 1,

characterized in

that the control means (2) are used for receiving incoming addresses, e.g. IP-addresses, and for initiating a look-up of the corresponding next-hop address, comprising a look-up in the cache memory (4;4';24;34) and if needed, a traversion of the digital search tree (5;5';25;35).

3. An arrangement according to claim 1 or 2,

characterized in

that a traversion of the digital search tree (5;5';25;35) is initiated substantially simultaneously with the look-up in the cache (4;4';24;34).

4. An arrangement according to claim 3,
characterized in
that the traversion of the digital search tree (5;5';25;35) is interrupted upon a cache hit.

5. An arrangement according to claim 1 or 2,
characterized in
that traversion of the digital tree (5;5';25;35) only is initiated if the look-up in the cache (4;4';24;34) is unsuccessful, i.e. if the address with an index to the next-hop address cannot be found.

6. An arrangement according to any one of the preceding claims,
characterized in
that indices to the leaves of a number of the most recently looked-up next-hop addresses are stored in the cache memory (4;4';24;34).

7. An arrangement according to any one of the preceding claims,
characterized in
that the least recently used addresses in the cache (4;4';24;34) are successively replaced by newly looked-up addresses.

8. An arrangement according to any one of claims 1-6,
characterized in
that a time interval is defined for replacement of addresses in the cache memory (4;4';24;34) such that addresses not having been used during such interval are replaced by new addresses.

9. An arrangement according to any one of the preceding claims,
characterized in
that the access time for a cache access is shorter than one digital search tree (trie) memory access.

10. An arrangement according to any one of the preceding claims, characterized in that the cache memory (4;4';24;34) comprises a direct map cache, a preset associative cache or a fully associative cache.

11. An arrangement according to any one of the preceding claims, characterized in that the arrangement is implemented as an ASIC chip.

12. An arrangement according to any one of claims 1-10, characterized in that the arrangement is implemented as FPGA chip.

13. An arrangement according to any one of the preceding claims, characterized in that the digital tree structure is provided on-chip.

14. An arrangement according to any one of claims 1-12, characterized in that the storing means implemented as a digital tree structure is provided off-chip.

15. An arrangement according to any one of the preceding claims, characterized in that the cache memory is provided on-chip.

16. An arrangement according to any one of the preceding claims, characterized in that the arrangement comprises an IP-(Internet) router.

17. An arrangement according to any one of the preceding claims, characterized in

that the routing table comprising a digital search tree is a compressed data structure.

18. An arrangement according to any one of the preceding claims, characterized in
that the digital tree comprises an LC-trie, a Patricia trie or any other convenient trie.

19. An arrangement according to any one of the preceding claims, characterized in
that a number of addresses are looked up in parallell.

20. An arrangement according to any one of the preceding claims, characterized in
that pipe-lining is implemented.

21. A method of routing a packet through a routing device of a packet switched data network, said routing device including a control means (2) and router storing means (3;3';23;33)comprising a digital search tree, the method being characterized through the steps of:

- storing indices to the next-hop addresses for a number of data packet addresses in an associative cache memory,
- using the cache-memory and a traversion of the digital search tree to find an index to the next-hop address or to the leaf containing the next-hop address,
- interrupting the traversion of the digital search tree in case of a cache hit, otherwise
- proceeding with the traversion of the digital search tree until the next-hop address is found,
- fetching the next-hop address information and forwarding the packet to the relevant output port.

22. The method of claim 21,
characterized in
that it comprises:

- first performing a look-up in the cache-memory; in case of no cache hit,
- initiating a traversion of the digital search tree.

23. The method of claim 21,
characterized in
that it comprises:

- simultaneously performing a look-up in the cache-memory and initiating a traversion of the digital search tree,
- interrupting the traversion of the digital search tree in case of a cache hit, otherwise
- proceeding with the traversion until the leaf containing the next-hop address is found.

24. The method of claim 22 or 23,
characterized in
that it comprises the step of, when a next-hop address is found through the traversion of the digital search tree,

- storing the index to said next-hop address into the cache memory, e.g. replacing the least recently accessed address information if there is no free space.

25. The method of claim 21,
characterized in
that it comprises the step of:

- replacing the index to the least recently accessed address information in the cache memory through the index to the most recently used address information.

26. The method according to any one of claims 21-25,

characterized in
that the cache-memory is a fully associative cache provided on-chip of an FPGA or an ASIC used for the implementation of the routing device.

27. The method of any one of claims 21-26,
characterized in
that a number of next-hop addresses are looked-up substantially in parallell.

28. A packet data communication system comprising a number of routing devices within a network or interconnecting networks, each routing device comprising a number of input and output ports, control means comprising a forwarding engine and router storing means including a digital search tree (a trie) containing next-hop address information,
characterized in
that in at least a number of routing devices a cache memory is provided for storing indices to a number of next-hop addresses contained in the digital search tree and in that for finding a next-hop-address a look-up is performed at least in the cache memory.

29. A system according to claim 28,
characterized in
that a traversion of the digital search tree is initiated substantially simultaneously with the look-up in the cache-memory and in that it is interrupted in case of a cache hit.

30. A system according to claim 28,
characterized in
that a traversion of the digital search tree only is initiated if there is no cache hit.

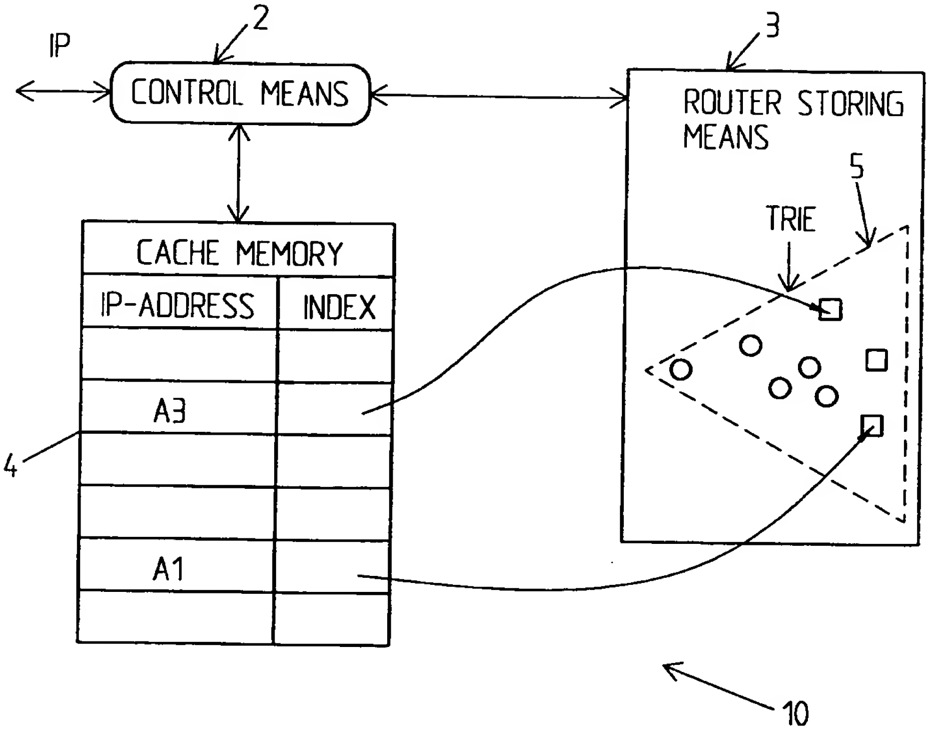

IP
2
CONTROL MEANS
3
ROUTER STORING MEANS
5
TRIE
CACHE MEMORY
IP-ADDRESS
INDEX
A3
4
A1
10

Fig. 1

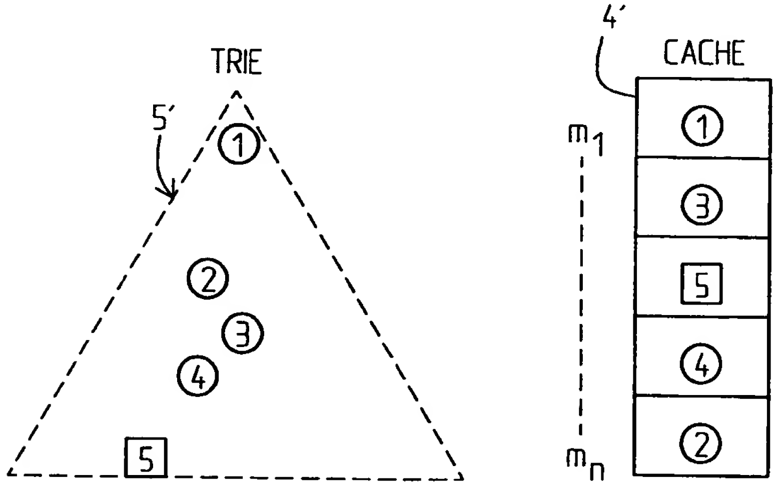

TRIE
5'
1
2
3
4
5
4'
CACHE
m_1
m_n
1
3
5
4
2

Fig. 2

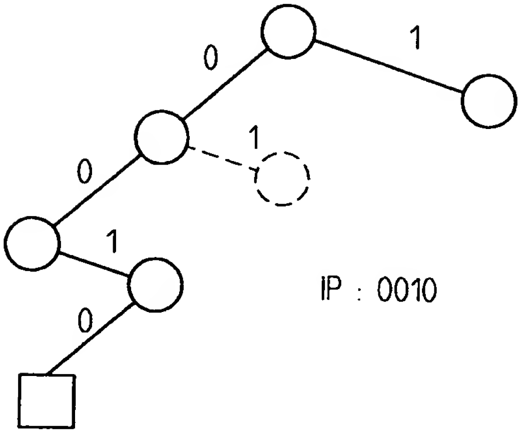

0
1
0
1
1
0
IP : 0010

Fig. 3

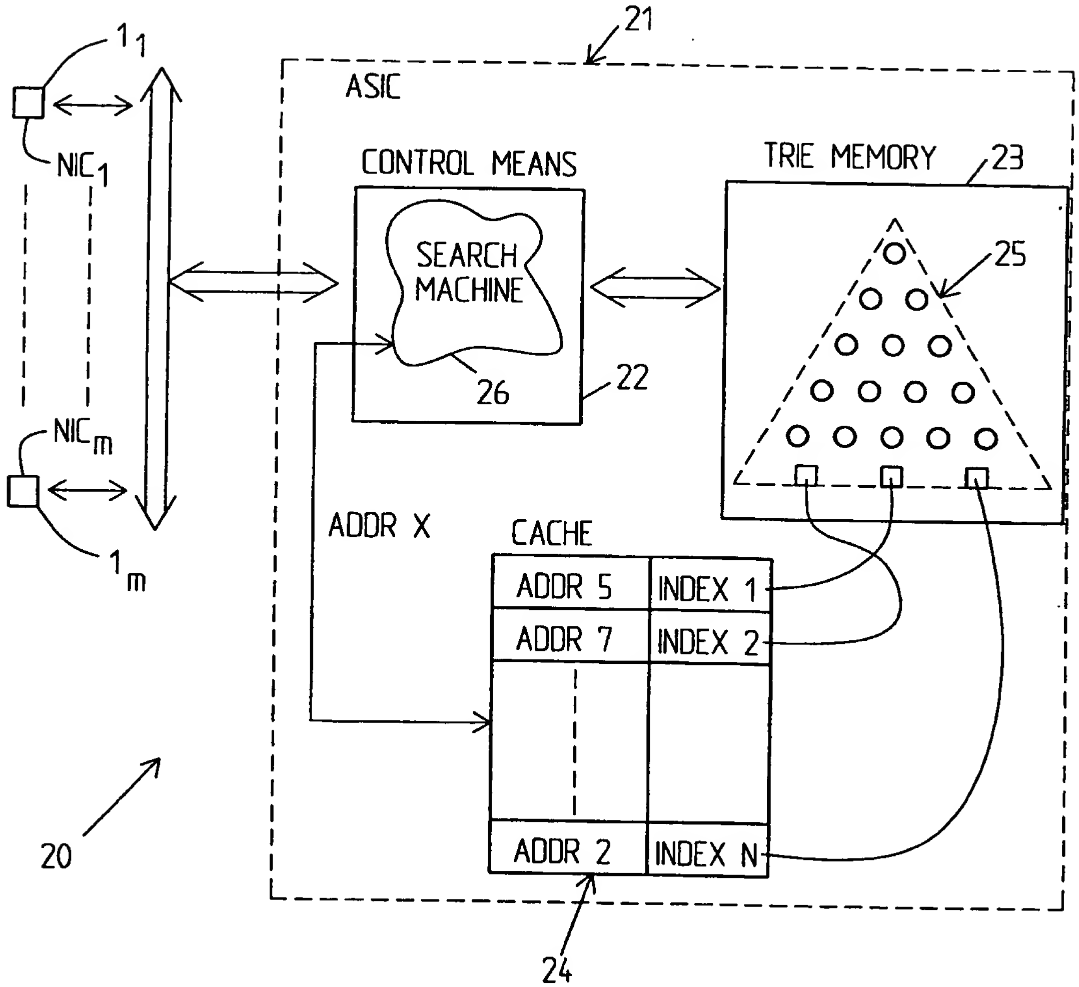

21
ASIC
1_1
NIC_1
NIC_m
1_m
CONTROL MEANS
SEARCH MACHINE
26
22
TRIE MEMORY
23
25
ADDR X
CACHE
ADDR 5
INDEX 1
ADDR 7
INDEX 2
ADDR 2
INDEX N
24
20

Fig. 4

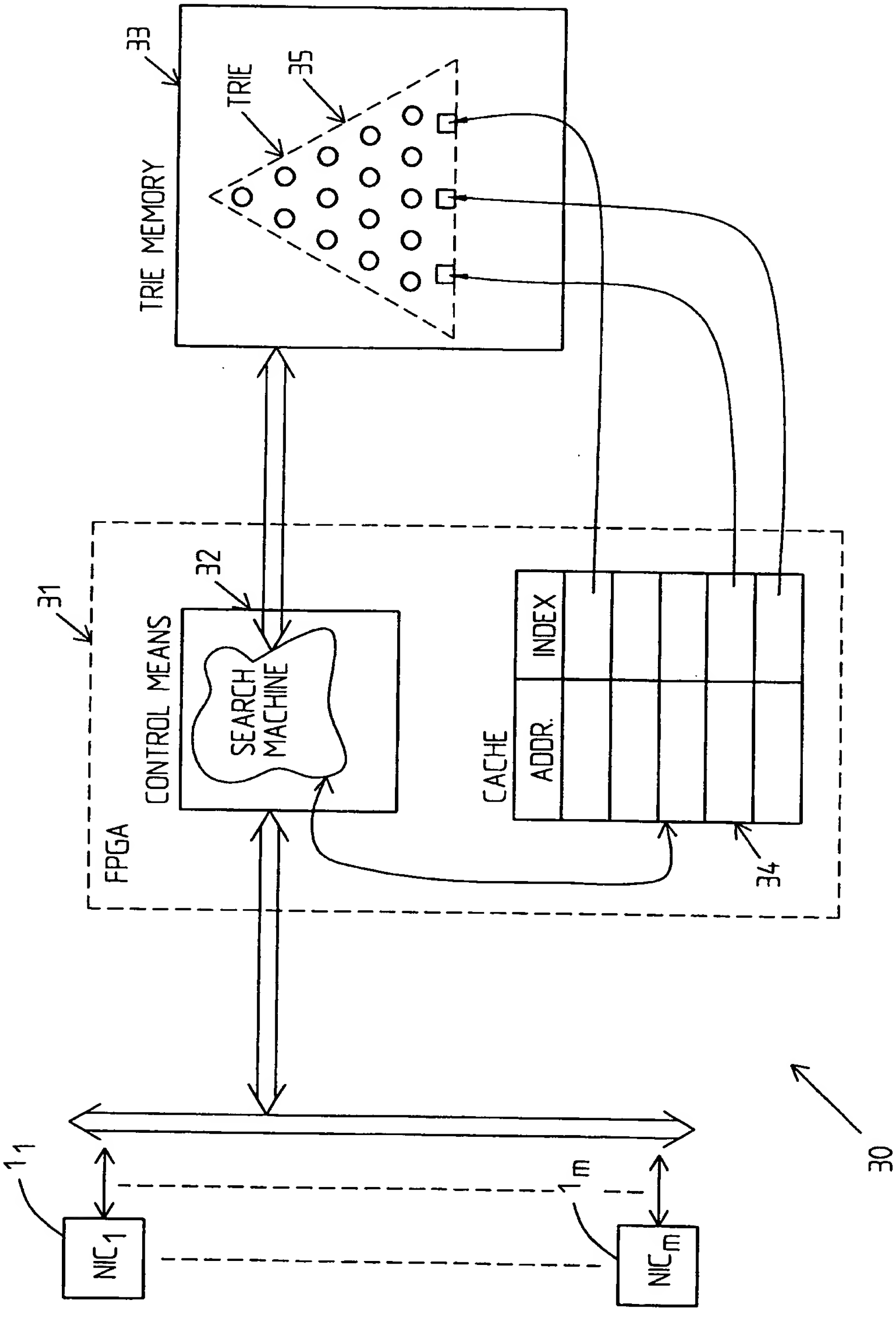

TRIE MEMORY
33
TRIE
35
CONTROL MEANS
32
SEARCH MACHINE
31
FPGA
CACHE
ADDR.
INDEX
34
NIC1
NICm
11
1m
30

Fig. 5

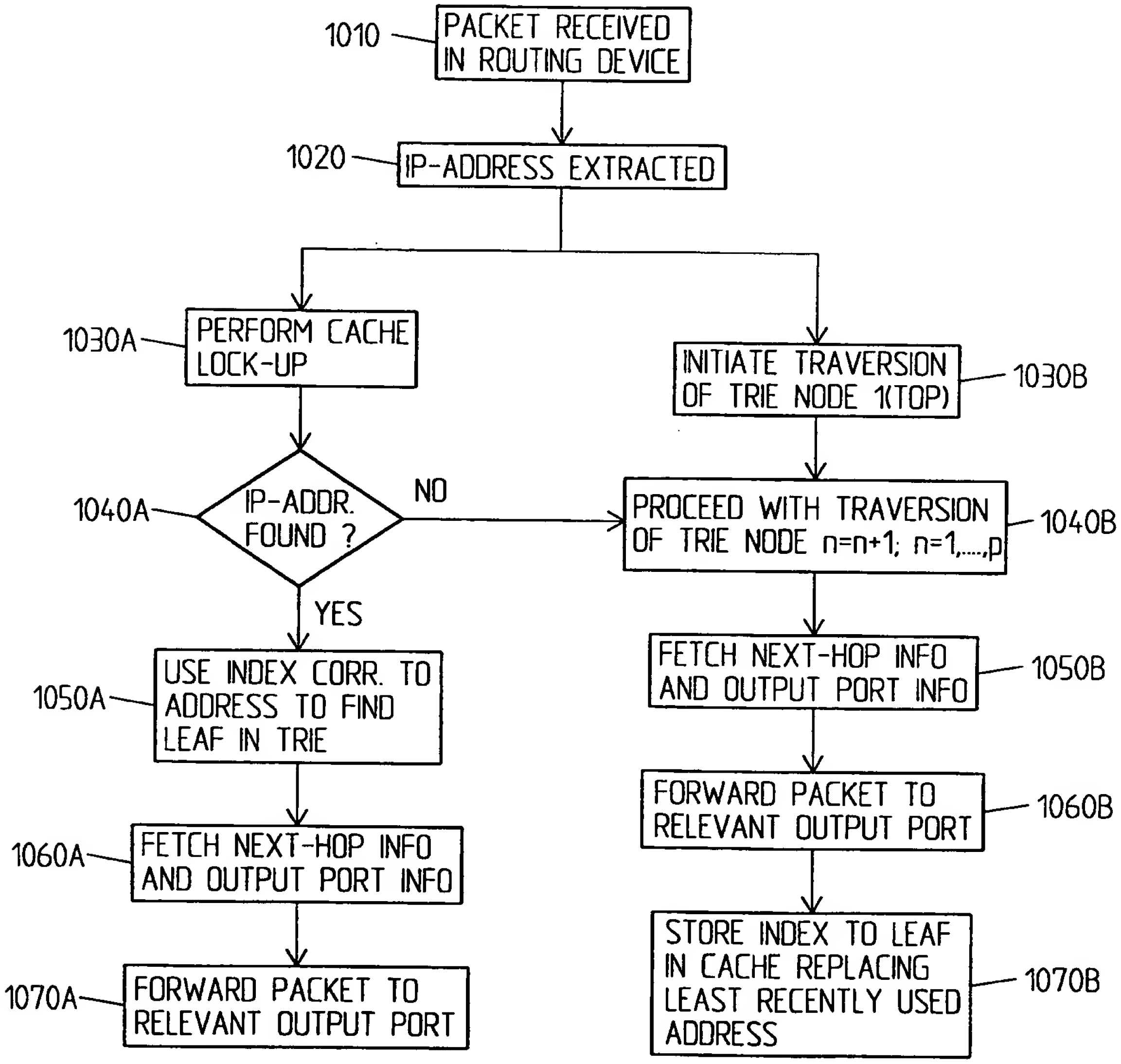

1010
PACKET RECEIVED IN ROUTING DEVICE
1020
IP-ADDRESS EXTRACTED
1030A
PERFORM CACHE LOCK-UP
1030B
INITIATE TRAVERSION OF TRIE NODE 1(TOP)
1040A
IP-ADDR. FOUND ?
NO
YES
1040B
PROCEED WITH TRAVERSION OF TRIE NODE n=n+1; n=1,....,p
1050A
USE INDEX CORR. TO ADDRESS TO FIND LEAF IN TRIE
1050B
FETCH NEXT-HOP INFO AND OUTPUT PORT INFO
1060A
FETCH NEXT-HOP INFO AND OUTPUT PORT INFO
1060B
FORWARD PACKET TO RELEVANT OUTPUT PORT
1070A
FORWARD PACKET TO RELEVANT OUTPUT PORT
1070B
STORE INDEX TO LEAF IN CACHE REPLACING LEAST RECENTLY USED ADDRESS

Fig. 6

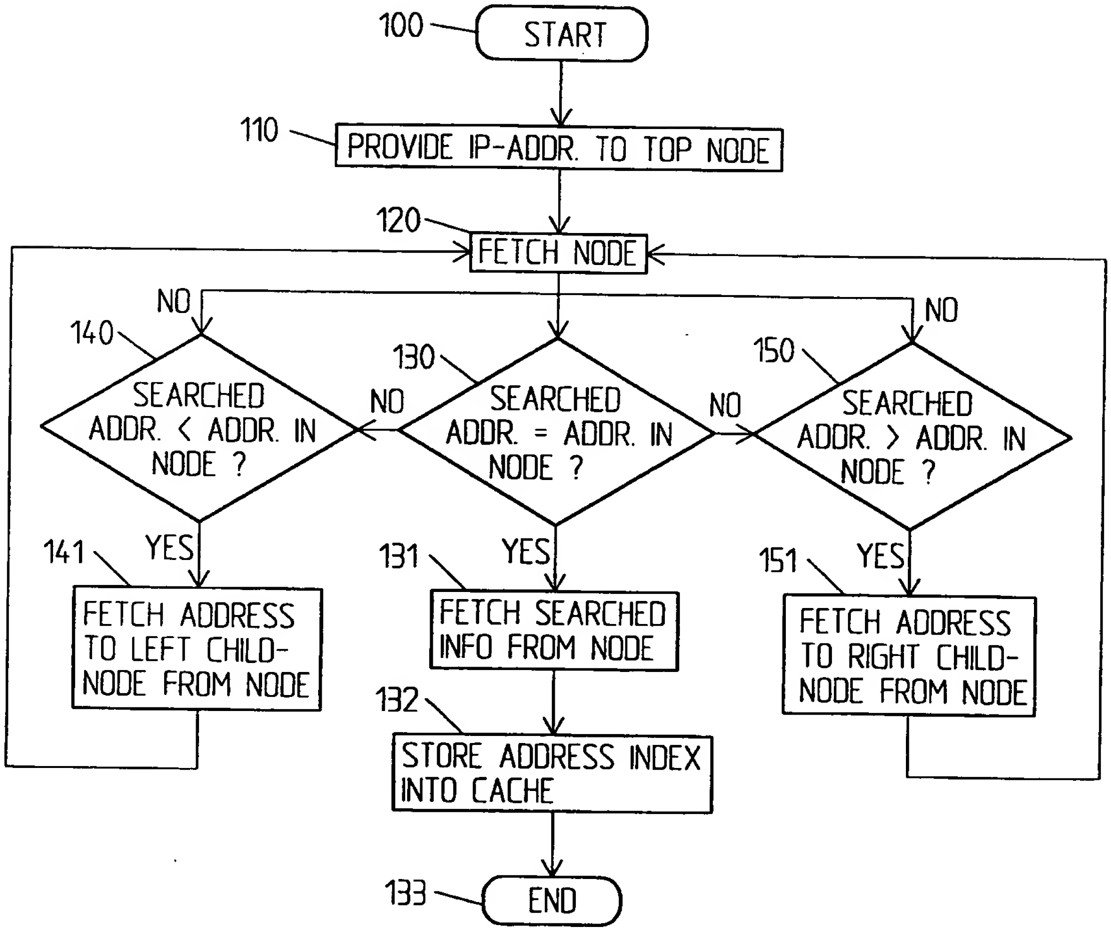

100
START
110
PROVIDE IP-ADDR. TO TOP NODE
120
FETCH NODE
NO
NO
140
SEARCHED ADDR. < ADDR. IN NODE ?
NO
130
SEARCHED ADDR. = ADDR. IN NODE ?
NO
150
SEARCHED ADDR. > ADDR. IN NODE ?
141
YES
FETCH ADDRESS TO LEFT CHILD-NODE FROM NODE
131
YES
FETCH SEARCHED INFO FROM NODE
151
YES
FETCH ADDRESS TO RIGHT CHILD-NODE FROM NODE
132
STORE ADDRESS INDEX INTO CACHE
133
END

Fig. 7

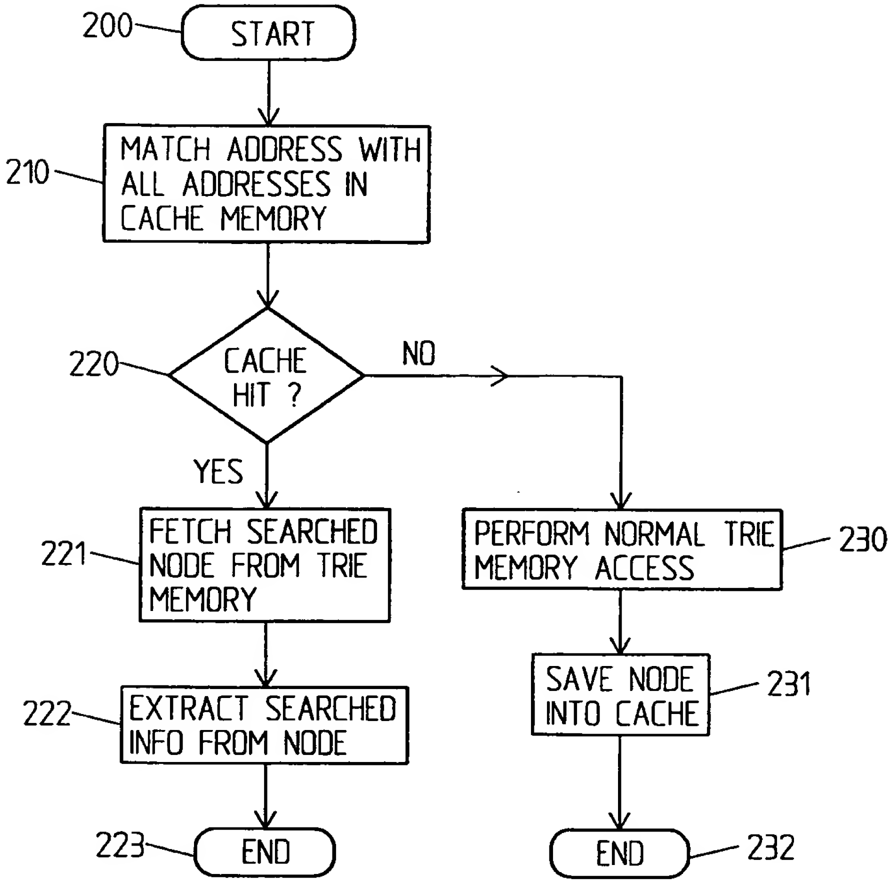

200
START
210
MATCH ADDRESS WITH ALL ADDRESSES IN CACHE MEMORY
220
CACHE HIT ?
NO
YES
221
FETCH SEARCHED NODE FROM TRIE MEMORY
230
PERFORM NORMAL TRIE MEMORY ACCESS
222
EXTRACT SEARCHED INFO FROM NODE
231
SAVE NODE INTO CACHE
223
END
232
END

Fig. 8

INTERNATIONAL SEARCH REPORT

International application No.
PCT/SE 00/01162

A. CLASSIFICATION OF SUBJECT MATTER

IPC7: H04L 12/56, G06F 17/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC7: H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE,DK,FI,NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P,A	US 6041053 A (JOHN R. DOUCEUR ET AL), 21 March 2000 (21.03.00), column 4, line 14 - column 6, line 27; column 15, line 10 - column 17, line 52 --	1-30
P,A	EP 0926922 A2 (NORTHERN TELECOM LIMITED), 23 December 1998 (23.12.98), abstract --	1-30
A	US 5640551 A (KE-CHIANG CHU ET AL), 17 June 1997 (17.06.97), column 2, line 46 - column 3, line 5, abstract --	1-30

☒ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" erlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search 8 Sept 2000	Date of mailing of the international search report 27 -09- 2000
Name and mailing address of the ISA Swedish Patent Office Box 5055, S-102 42 STOCKHOLM Facsimile No. +46 8 666 02 86	Authorized officer Kristoffer Ogebjer/MN Telephone No. +46 8 782 25 00

INTERNATIONAL SEARCH REPORT

International application No.
PCT/SE 00/01162

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5546390 A (GEOFFREY C. STONE), 13 August 1996 (13.08.96), abstract --	1-30
A	US 5227778 A (GARY V. VACON ET AL), 13 July 1993 (13.07.93), abstract -- --------	1-30

INTERNATIONAL SEARCH REPORT

Information on patent family members

08/05/00

International application No.
PCT/SE 00/01162

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 6041053 A	21/03/00	NONE	
EP 0926922 A2	23/12/98	NONE	
US 5640551 A	17/06/97	NONE	
US 5546390 A	13/08/96	NONE	
US 5227778 A	13/07/93	NONE	